



Real-Time Human Detection and Tracking Based on Deep Learning Technique

2084

Gona M Rozhbayani^{1*}, Amel Tuama², Fadwa Al-Azzo³

Abstract

In the field of computer vision, the detection of an object such as a human is critical for image understanding. Human tracking detection in real-time helps in providing critical information for a vast variety of intelligent system applications. This paper presents a new model for real-time human tracking detection (RTHTD) for surveillance video by using a deep learning technique based on the modified YOLOv5 model, the backbone of the modified model formed from Cross Stage Partial(CSP), Bottleneck, and SPPF. To evaluate the proposed network results utilizing two different custom datasets. Low resolution, indoor, outdoor, diverse outfits, gender, occlusion, and other tough recording settings are all included in the datasets. In the training process of the proposed model gained confidence scores detection quality (total mAP) for the first dataset is 98.3%, and for the second dataset is 96.3% as well. The proposed system has been tested in real-time video frames with a high accuracy detection. The working environment for training the system is Google Colab with Tesla K80 and experiments have been executed on Windows 10, 64-bit, Intel(R) Core(TM) i7-3540H CPU @3.00 GHz, GeForce 820M GPU.

KeyWords: Human tracking detection, YOLOv5, deep learning, real-time, surveillance video.

DOI Number: 10.14704/nq.2022.20.6.NQ22202

NeuroQuantology 2022; 20(6):2084-2093

1. Introduction

In the last few years, fields like image analysis and video analysis have obtained an enormous number of applications. Computer vision and artificial intelligence are two major technologies commanding technological society. Computer vision is a technique and a part of artificial intelligence that trains computers to interpret, identify and locate objects in an image or video for visibility world like the world of humans. Human detection is the task of locating all instances of human beings present in an image, and it has been most widely accomplished by searching all locations in the image, at all possible scales, and comparing a small area at each location with known templates or patterns of people.

Furthermore, human tracking is the process of temporally associating the human detections within a video sequence to generate persistent paths, or trajectories, of the people.

However, the importance of human tracking detection in images, videos, and real-time in the computer vision field is growing day by day. It is being used in a wide range of applications, such as video surveillance (Apoorva et al., 2018), counting people in special places (Chan & Vasconcelos, 2012), and abnormal event detection (Sharma & Varshney, 2020). Human tracking detection is more challenging than tracking other objects because everyone is different in appearances, poses, scales, and sizes. Other challenges can include lighting status, camera position, and rapid changes in human movements.

Corresponding author: Gona M Rozhbayani^{1*}

Address: ^{1*}Department of Computer Engineering, Technical Engineering College Mosul, Northern Technical University, Mosul, Iraq;

²Lecturer, Department of Computer Engineering, Kirkuk Technical College Engineering, Northern Technical University, Kirkuk,

Iraq; ³Lecturer, Department of Computer Engineering, Technical Engineering College Mosul, Northern Technical University, Mosul, Iraq.

E-mail: ^{1*}gona.mohammed@ntu.edu.iq; ²amel.tuama@ntu.edu.iq; ³fadwaalezzo@ntu.edu.iq

Relevant conflict of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: Accepted:



Deep learning (DL) has lately been required to manage a variety of challenging essential issues. Generally, DL enables quick and precise analysis and comparison of massive amounts of data. Therefore, many researchers have proposed various methods for human detection and tracking under specific conditions. As a result, we are proposing a modified YOLOv5 model in this paper that would support human tracking detection in different environments as well as different camera positions facilely and precisely in real-time videos. The proposed system of the modified YOLOv5 custom architecture consists of Cross Stage Partial (CSP) Bottleneck along with three convolutions in the model's backbone and Sigmoid Linear Unit (SiLU) activation function. The model was trained on different custom datasets, the network modification was applied to gain a higher confidence score in real-time human tracking detection videos. The model implemented by the PyTorch framework is based on the Python programming language.

The remainder of this paper is structured as follows. Section 2 introduces the related works and highlights of the previous studies. In Section 3, we present the method of YOLOv5 and the proposed system in detail. In section 4 we give the experimental details and analyze the experimental results in datasets. Section 5 concludes this paper.

2. Related Work

Various approaches have been utilized for human detection and tracking in video. The (Chandan G et al., 2018) proposed a combination between Single Shot Detector (SSD) and MobileNets by deep learning to perform real-time object detection and tracking. This approach carried out effective object detection while not compromising the performance. Applications of computer vision like vehicle navigation and traffic surveillance video discover object detection and tracking as serious tasks. Thereafter, (Ravish Aradhya, 2019) proposed Convolution Neural Network (CNN) for single object detection and the YOLOv3 model for multiple object detection on KITTI and COCO datasets. The paper provided real-time, accurate, precise identifications appropriate for real-time traffic applications. Additionally, in the same year, Bhuyain Mobarok focused on providing security by monitoring crowded places using video surveillance systems. In this work (Hossain, 2019), a different approach for automatic human detection and tracking is proposed

by combining Histogram of Oriented Gradients (HOG) feature detection with a particle filter for tracking.

The system achieved more than 90% accuracy in human detection. However, for tracking objects over the frames YOLOv3 and Simple Online Real-Time Tracking (SORT) is used in (Ahmad et al., 2020). Human appearances are significant challenges for tracking, which affected the location of the person about the camera position. In this paper, an overhead camera is supplied for overhead human tracking by using Generic Object Tracking Using Regression Networks (GOTURN) after using (Faster-RCNN) for human detection in video sequences. Recently, in (Sruthi et al., 2021) presented a system utilizing the YOLOv5 model that offers to monitor isolated places in real-time and automatically detects stranded humans during floods. The model is trained using a particular dataset including more than 1000 images and got 0.954 mAP.

3. Method

3.1 YOLOv5 Model architecture

YOLOv5 named You Look Only Once (YOLO) is the most basic version of the object detection approach (Wu et al., 2021). By performing a Convolution Neural Network (CNN) architecture on the image, the position and class of the objects in the image are defined. YOLOv5 comes in four different models: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, each with a varied detection accuracy and performance. The multiplier of depth controls the model to determine YOLOv5, for instance, YOLOv5l has a depth multiple of 1, whereas YOLOv5s has a depth multiple of 0.33. YOLOv5s is the easiest version with the little weight file and has an elevated detection speed.

In the architecture, YOLOv5 is made up of three main parts: backbone, neck, and head as follows:

Backbone: is mainly used to extract significant features from the given input image. The backbone structure of YOLOv5 consists of some blocks, including Convolution, Convolution Three (C3), and Spatial Pyramid Pooling (SPP) (He et al., 2014). The Convolution structure is the smallest component in the YOLO network structure shown in figure 1, which processes the incoming image with a resolution of $640 \times 640 \times 3$. It initially turns into a $320 \times 320 \times 12$ feature map via the slicing operation, afterward a convolution operation of 32 convolution kernels, and



eventually turns into a $320 \times 320 \times 32$ feature map.

The convolution block involves a particular structural diagram made up of a convolution layer, a batch normalization (BN) layer, and a Sigmoid Linear Unit (SiLU) as an activation function layer. The C3 block contains CSP Bottleneck, with three convolutions shown in figure 2. The SPP block is the final layer of the backbone network of YOLOv5s architecture, its structure includes a convolutional with three parallel maximum pooling layers 5×5 , 9×9 , and 13×13 the resulting characteristics are concerted together to perform 275 characteristics after that convolutional is performed again. Figure 4. explains the YOLOv5s model architecture with SPP block.

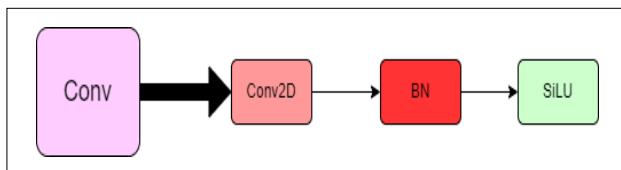


Figure 1. Convolutional block structure

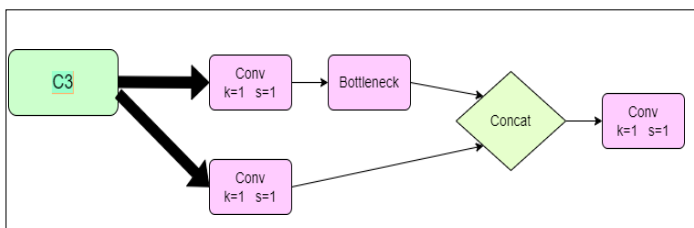


Figure 2. C3 block structure

Neck: this part of the model is utilized to generate feature pyramids. Feature pyramids help the model to generalize well on object scaling and perform great on unseen data. It helps to identify the same object with different sizes and scales. Other models use different types of feature pyramid techniques like FPN, PANet, etc. As the image passes through the various layers of the neural network, the complexity of the features **increases**, and the spatial resolution of the image **decreases** concurrently. Due to this, the pixel-level masks cannot be identified accurately by the high-level features.

Head: is always required to perform the final prediction part. It is composed of three convolution blocks, with vectors holding the estimated coordinates of the bounding box and confidence score, as well as the label. The output processing is completed by discarding boxes with poor scores.

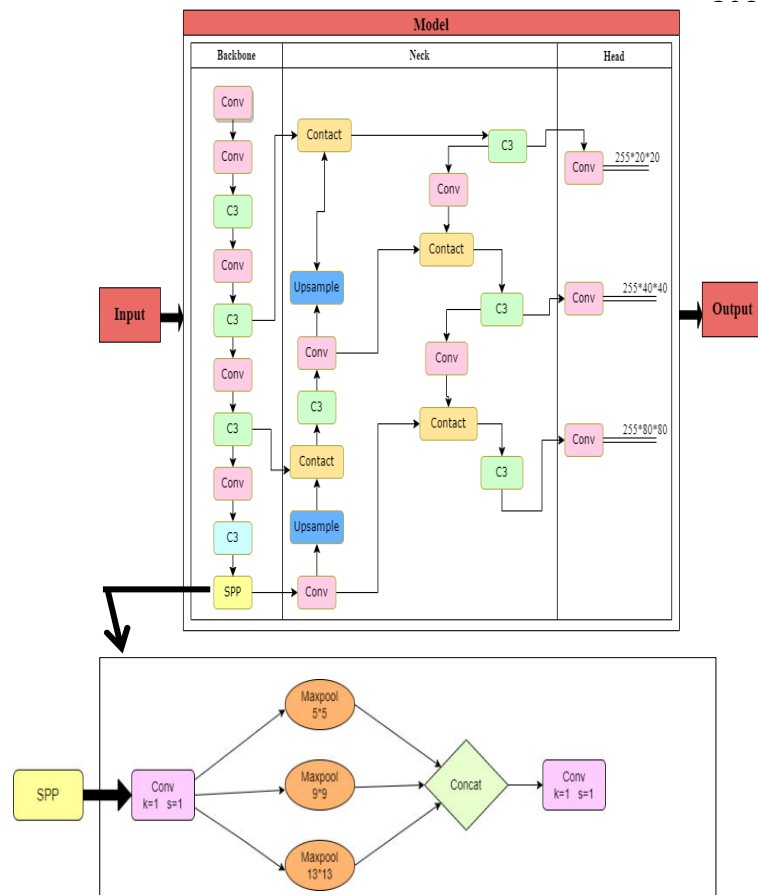


Figure 4. YOLOv5s model and SPP block structures

3.2 YOLOv5 Model Modification

Figure 4. illustrated the modified network of YOLOv5s which has been applied in this work, composed of the Spatial Pyramid Pooling Fast (SPPF) block by Glenn Jocher and laid the C3 block that consists of CSP Bottleneck modules after the SPPF block.

The SPPF block was applied instead of SPP for gaining double speed of network performance and high accuracy in the real-time human tracking detection process. SPPF utilizes three individual 5×5 maximum poolings in place of the original 5×5 , 9×9 , and 13×13 , the input serially passes through the multiple 5×5 maximum pooling which reduces computation to speed up. However, the SPP block includes three parallel maximum pooling layers the feature maps of different scales are analyzed separately then the concat operation is applied.

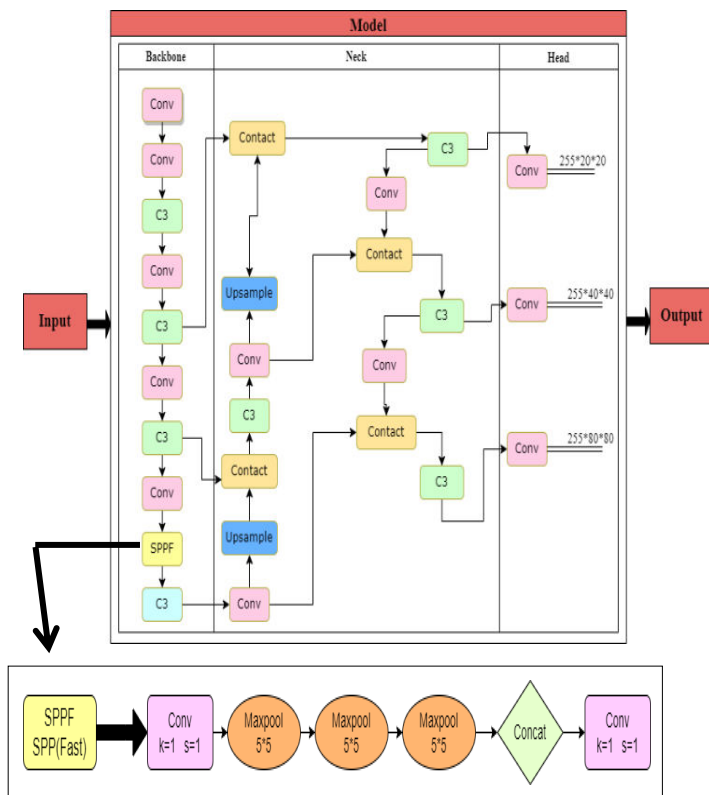


Figure 4. YOLOv5s model modification and SPPF block structures

3.3 Proposed System

The proposed system of this paper shown in figure 5, aims to detect and track humans in real-time videos with different custom datasets in higher mAP scores during the training process, along with a more accurate confidence score in detection. The first step is to label the dataset by drawing a bounding box BBox (ground truth) around all humans regardless of other objects and background in the image. Then preprocessing is applied to the dataset by resizing each image to speed up the training process of the network. Furthermore, the data augmentation technique is applied as well, to improve model robustness. The process of training is done by modifying YOLOv5 on different custom datasets for obtaining the weight file ".pt" from the model. Eventually, deploying the model for the prediction of real-time human tracking detection.

The most commonly used evaluation metrics for checking the model performance are precision, recall, and mAP. The number of correct class expectations

that correspond to the positive class is defined as precision. On the other hand, recall is a measure of how many positive class expectations were formed from all positive instances in the dataset. Mean Average Precision (mAP) is a metric used for evaluating object detection models, values are calculated from 0 to 1.

2087

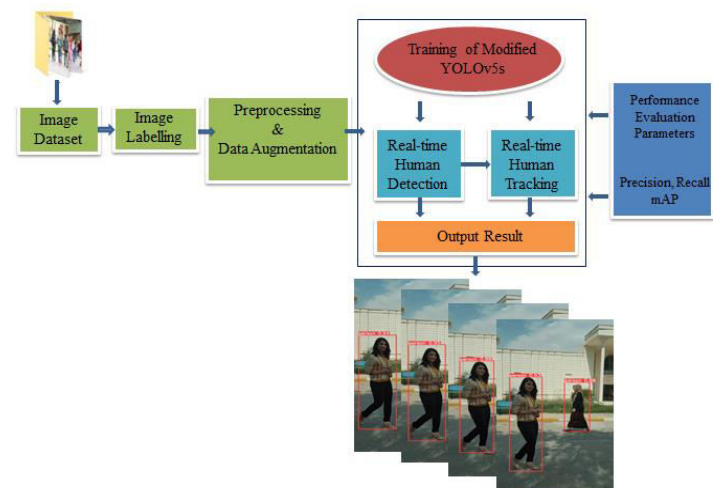


Figure 5. The block diagram of the proposed RHTD system

3.4 Dataset and Preprocessing

The proposed RHTD system is evaluated on two different custom datasets as shown in Figure 6,7. The first dataset called Open Images Figures (OID)(Kuznetsova et al., 2018) composed of 517 images, while the second dataset collected from Kirkuk Technical Engineering College (KTEC)(Kirkuk Technical College Engineering, 1998) contains 650 images. The datasets contain different challenging recording conditions such as low resolution, indoor, outdoor, various outfits,gender, occlusion, etc. As well the dataset preprocessing is obtained to modify and resize each image to 640*640 for improving the training efficiency model. Then the datasets were split into a training set and validation set of 80% and 20% respectively to evaluate the candidate model efficiency performance. The training dataset was created by manually tagging the human image using Roboflow software (Joseph and Brad, 2019) which is a computer vision developer framework and a type of computer vision annotation tool (CVAT) because it is required to have a ground truth of what precisely the object (person) is. In other words, it is essential to draw a bounding box (BBox) around only the person regardless of the rest of the other objects in the same image as shown in figure 8. Therefore the proposed system understands that this "person" inside the box



belonged to the person class. Roboflow saves the annotations as text file ".txt" format per image. The format is represented as < object-class-ID><X center><Y center><Box width><Box height>, each text file contains one BBox annotation for each of humans in the image, one row per human. Afterward, all ".txt" files are collected in a labels folder.



Figure 6. Samples of the OID dataset



Figure 7. Samples of the KTEC dataset

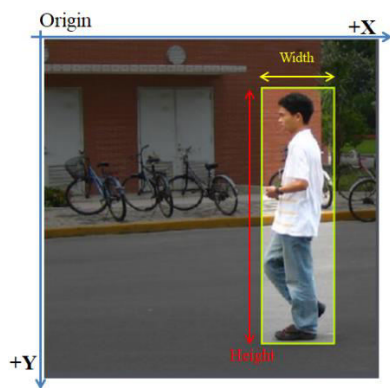


Figure 8. Sample of person class Surrounded by a Bounding Box.

4. Experimental Results

4.1 Experimental Environment

PyTorch is an open-source machine learning (ML) framework based on the Torch library and the Python programming language (Paszke et al., 2019), also it is one of the most advanced deep learning research platforms. PyTorch is designed to execute operators asynchronously on a Graphics Processing Unit (GPU), which allows the model to execute Python code on the CPU and GPU. The framework is designed to accelerate the transition from research prototype to deployment.

Training the model on custom datasets on Google Colab in PyTorch framework, which offers free access to Google computing resources such as (GPU) for training the model. Eventually, deploying the model for the prediction of real-time human tracking detection.

The processor is Intel(R) Core(TM) i7-3540H CPU

@3.00 GHz, memory 8.00 GB, GPU is GeForce 820M. The operating system is Windows 10, 64-bit, CUDA 11.3, and cuDNN 8.4.0. The deep learning framework is PyTorch 1.11.0 + cu113, python version 3.9.5. In addition to training the neural networks, Google Colab with Tesla K80 graphics cards is used too. It offers both free and paid Google cloud computing services that can be used in a variety of computer applications.

4.2 Evaluation metrics and losses

4.2.1 Evaluation metrics: Precision, recall, and mAP are the most important evaluation metrics for such implementation. They are used to evaluate the accuracy of the model.

Precision is the proportion between the number of accurately classified positive answers (true positive) and the total of correct positive answers (true positives) and incorrect but positively labeled answers (false positive), (Equation 1).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

Where TP = true positive, FP = false positive.

Recall: calculates the number of positive labels that can be correctly predicted out of all positive labels. It is the proportion of accurate positive answers (true positives) to the total of correct positive answers (true positives) and incorrectly negatively labeled answers (false negative). (Equation 2)

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

Where FN = false negative.

Mean Average Precision (mAP): The mAP is calculated by finding the Average Precision (AP) for each class and then averaging it over several classes. (Equation 3).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3)$$

AP_i = AP of class i

N = number of classes

AP: on detecting bounding boxes, the AP is the consequence of precision and recall. The AP score is defined as the mean precision at the set of 11 equally spaced recall values, $\text{Recall}_i = [0, 0.1, 0.2, \dots, 1.0]$. Thus, (Equation 4).

$$AP = \frac{1}{11} \sum_{\text{Recall}_i} \text{Precision}(\text{Recall}_i) \quad (4)$$

The precision at recall_i is taken to be the maximum precision measured at a recall exceeding Recall_i .

4.2.2 Evaluation losses: The three sections of the YOLO loss function are:

1. Coordinate or box loss (box_loss) as a result of incorrectly bounding box placement.
2. Objectness loss (obj_loss) is caused by incorrect box-object prediction. Besides, it is Binary Cross Entropy which means the measurement output is a probability value between 0 and 1. '0' indicates the wrong classes for the object in that box and '1' for the correct class.
3. Classification loss (cls_loss) is the squared error of the class conditional probabilities for each class, it is cross-entropy as well.

4.3 Experimental Training Results

The training procedure train the model to produce the desired results by finding a cluster of weights in the network for handling particular issues. In this work, the pre-trained YOLOv5s model and modified YOLOv5s were fine-tuned for two custom datasets to establish human tracking detection to achieve the best accuracy of the network performance. A default dataset of the YOLOv5 model contains 80 different classes with COCO128 equals 0.372 mAP (Glenn Jocher, 2022).

Within the proposed model worthy features are extracted in the backbone part. Moreover, Path Aggregation Network (PANet) (Liu et al., 2018) is used as a neck to collect feature pyramids. PANet takes a bottom-up path to a top-down path to extract and combine semantically privileged features with precise localization information. The latter part of the model applied anchor boxes on features and generates final output vectors with class probabilities, objectness scores, and bounding boxes. For training the OID and KTEC datasets, different epoch numbers tried to get the best evaluation metric values. As the number of epochs increases, the values of evaluation metrics are changed goes from optimal to overfitting. The more preferable result and highest accuracy were on 200 epochs, as shown in Tables 1 and 2, respectively. In the training of the YOLOv5s model using the PyTorch framework in the Google Colab environment, the batch size refers to the number of samples chosen for each training set equal to 16, and the image size is (640*640*3). The hyperparameters whose values control the whole learning process of the model are learning-rate start (lr0) equal to 0.1, it determines step size and training progress will increase by lr0 value, and momentum is 0.937, which is the tuning parameter for the gradient descent algorithm, it is responsible for replacing the gradient with an aggregate of the gradient.

The OID and KTEC training process progress of Precision, Recall, and mAP values are shown in figures (9,10), respectively. The benchmark is a scale from 0 to 1 that measures the output value at Y-axis, and X-axis indicates the epoch number. Results of the OID and KTEC evaluation losses box_loss, obj_loss, and cls_loss are shown in figures (11,12), respectively. This work, only contains one class, there are no class misidentifications, and the classification error is always zero.

Table1. Experimental results of evaluation metrics for the OID dataset with different epoch numbers.

Epochs number	Precision	Recall	mAP@0.5
50	0.965	0.918	0.98
100	0.967	0.951	0.983
150	0.894	0.967	0.979
200	0.967	0.953	0.981

Table2. Experimental results of evaluation metrics for the KTEC dataset with different epoch numbers.

Epochs number	Precision	Recall	mAP@0.5
50	0.949	0.904	0.938
100	0.962	0.916	0.937
150	0.986	0.904	0.963
200	0.938	0.916	0.946

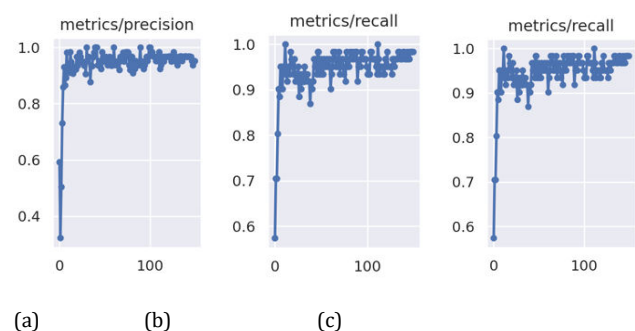


Figure 9. Results of metrics in training progress of the OID dataset. (a) Precision. (b) Recall. (c) mAP.



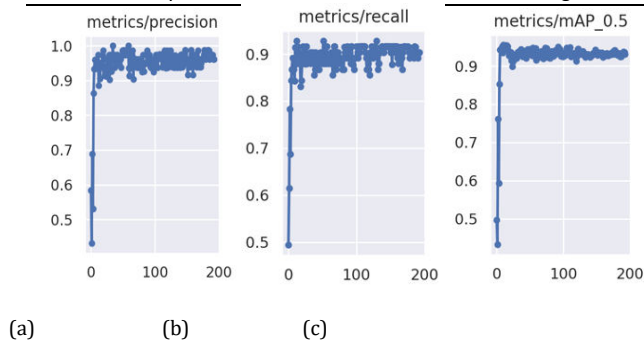


Figure 10.Results of metrics in training progress of the KTEC dataset. (a) Precision. (b) Recall. (c) mAP.

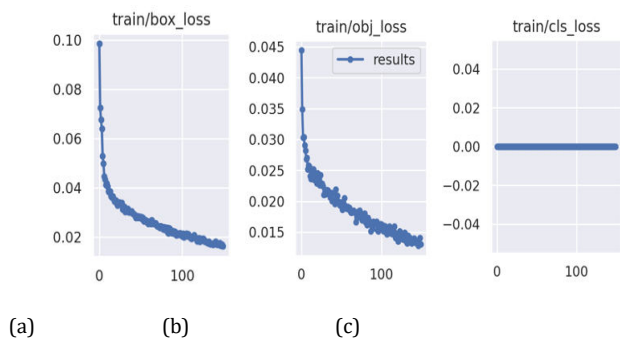


Figure 11.Results of losses in training progress of the OID dataset. (a) box_loss. (b) obj_loss. (c) cls_loss.

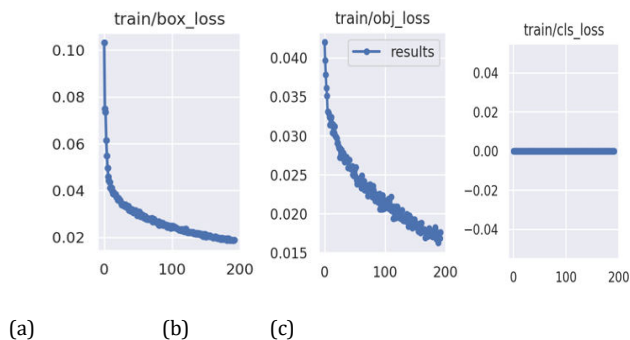


Figure 12.Results of losses in training progress of the KTEC dataset. (a) box_loss. (b) obj_loss. (c) cls_loss.

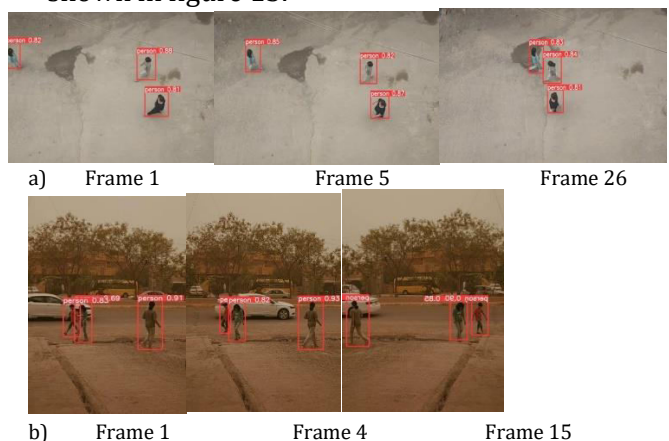
As raising feature maps of the dataset confidence score on real-time testing increases as well. The OID and KTEC datasets are merged to bring out and provide the third dataset to become 1167 images. As a result of training the third dataset with the default, YOLOv5s network architecture was 0.905 mAP. On the other hand, the same dataset trained on the proposed modified YOLOv5s model to gain more accurate results, and obtained accuracy was increased to 0.913 mAP along with a more elevated confidence score. Table 3 illustrates the comparison between the default and modified YOLOv5s model of evaluation metrics results.

Table3.Experimental results of evaluation metrics comparison between default and modified YOLOv5s model.

Model	Precision	Recall	mAP@0.5
Default YOLOv5s	0.872	0.835	0.905
Modified YOLOv5s	0.923	0.84	0.913

4.4 Experimental Testing Results

The proposed network performance was tested on realistic real-time video frames from different camera positions, and in diverse scenarios comprising indoor, outdoor, midnight, dusty environments, and overview scenes. The testing process is accomplished by setting a bounded box with the confidence score on top of it, despite the diverse appearances, gender, and shape. To properly illustrate the findings of the trained proposed system, there are some frames to visualize the results shown in figure 13.



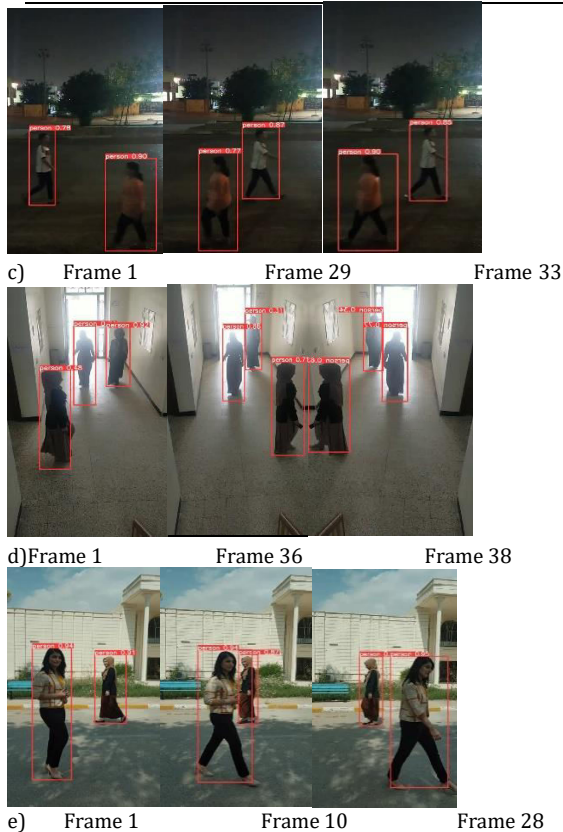


Figure 13.Results of human tracking detection with the proposed model. (a) Overview, (b) Dusty weather, (c) Night Day, (d) Indoor, (e) Outdoor.

As reflected in figure 13, the RTHTDsystem in this paper has been successful in detecting and tracking humans under a variety of camera settings. This also indicates the effectiveness and robustness of the proposed model performance for human tracking detection in real-time. In case the camera is mounted around 8 meters far away from the ground and the humans appear as a small shape as indicated in figure13(a), the system's efficiency was good at the detection accuracy reached more than 88%. Besides, the RTHTDsystem was tested in foggy and dusty environmental conditions and crowded backgrounds with different objects. Although the human visibility decreases due to low illumination in dusty weather and at night as shown in figure13(b,c), the detection accuracy results of the RTHTDsystem were satisfying more than 91% and 90% respectively. In indoor and outdoor places as shown in figure13(d,e), the system performance efficiency established was 94% and 92% respectively. The speed of the prediction processes time in this work using Intel(R) Core(TM) i7-3540H CPU processor was around 1ms to 1.7ms, while the inference time ranged from 300ms to 400ms at the confidence threshold value

equals 0.25, and the 30 FPS outcome from the webcam in real-time.

5. Conclusion

This paper introduces a real-time computer vision system for human tracking detection (RTHTD) based on the YOLOv5s model with varied two custom datasets, including the structure YOLOv5s network and dataset processing. The model is trained by two custom datasets OID and KTEC with the default YOLOv5s architecture for detection in real-time. After that, the two datasets are merged, as increasing feature maps of the dataset confidence score on real-time testing gains as well. Then the model architecture is modified by applying the Spatial Pyramid Pooling Fast (SPPF) block and laying the C3 block after the SPPF block. The modified YOLOv5s model fulfilled a higher mAP, along with a greater confidence score in testing real-time human detection tracking.

Acknowledgments

We would like to thank Kirkuk Technical Engineering College (Northern Technical University) for enabling us to use their archive in creating the KTEC dataset.

References

- Adhiparasakthi Engineering College. Department of Electronics and Communication Engineering, Institute of Electrical and Electronics Engineers. Madras Section & Institute of Electrical and Electronics Engineers. (n.d.). *Proceedings of the 2018 IEEE International Conference on Communication and Signal Processing (ICCSP) : 3rd - 5th April 2018, Melmaruvathur, India.*
- Ahmad, M., Ahmed, I., Khan, F. A., Qayum, F. & Aljuaid, H. (2020). Convolutional neural network-based person tracking using overhead views. *International Journal of Distributed Sensor Networks*, 16(6).
<https://doi.org/10.1177/1550147720934738>
- Chan, A. B. & Vasconcelos, N. (2012). Counting people with low-level features and bayesian regression. *IEEE Transactions on Image Processing*, 21(4), 2160–2177.
<https://doi.org/10.1109/TIP.2011.2172800>
- He, K., Zhang, X., Ren, S. & Sun, J. (2014). *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.*
https://doi.org/10.1007/978-3-319-10578-9_23
- Hossain, B. M. (2019). *Automatic Human Detection and Tracking in Crowded Scenes using Histogram of Oriented Gradients (HOG) and Particle Filter.*
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., Duerig, T. & Ferrari, V. (2018). *The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale.*
<https://doi.org/10.1007/s11263-020-01316-z>
- Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J. (n.d.). *Path Aggregation Network for Instance Segmentation.*
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury Google, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Xamla, A. K., Yang, E., Devito, Z., Raison Nabla, M., Tejani, A., Chilamkurthy, S., Ai, Q., Steiner, B., ... Chintala,



- S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*.
- Ravish Aradhya, H. (2019). Object Detection and Tracking using Deep Learning and Artificial Intelligence for Video Surveillance Applications. In *IJACSA International Journal of Advanced Computer Science and Applications* (Vol. 10, Issue 12). www.ijacsa.thesai.org
- RVS College of Engineering & Technology & Institute of Electrical and Electronics Engineers. (n.d.). *Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018) : date: July 11-12, 2018*.
- Sharma, A. & Varshney, N. (n.d.). Identification and Detection of Abnormal Human Activities using Deep Learning Techniques. In *European Journal of Molecular & Clinical Medicine* (Vol. 7).
- Sruthi, M. S., Poovathingal, M. J., Nandana, V. N., Lakshmi, S., Samshad, M. & Sudeesh, V. S. (2021). YOLOv5 based Open-Source UAV for Human Detection during Search And Rescue (SAR). *2021 International Conference on Advances in Computing and Communications (ICACC)*, 1–6.
- Wu, T. H., Wang, T. W. & Liu, Y. Q. (2021). Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network. *2021 3rd World Symposium on Artificial Intelligence, WSAI 2021*, 24–28.
<https://doi.org/10.1109/WSAI51899.2021.9486316>
https://www.facebook.com/TechnicalCollege.Kirkuk.NTU/photos/?ref=page_internal
<https://blog.roboflow.com/getting-started-with-roboflow/>
https://wandb.ai/glenn-jocher/yolov5_tutorial/reports/YOLOv5-COCO128-Tutorial-Results--VmlldzoZMDI5OTY



